

WHAT IS CLAIMED IS:

1. A method of transforming bytecode, the method comprising:  
determining an abstract bytecode by performing a compilation of an application for execution on a virtual device;  
transmitting the abstract bytecode from a service peer to at least a client peer;  
transforming the received abstract bytecode into a native bytecode for a client specific device that is connected to the client peer, wherein the client specific device is configurable;  
configuring the client specific device using at least in part the native bytecode;  
and  
executing the native bytecode at the client peer on the client specific device.
2. The method of Claim 1, wherein the abstract bytecode comprises abstract hardware bytecode that includes configuration information of a virtual device.
3. The method of Claim 1, wherein the abstract bytecode comprises abstract hardware bytecode and abstract software bytecode.
4. The method of Claim 1, wherein the abstract bytecode comprises abstract software bytecode and wherein transforming transforms the abstract software bytecode into native software bytecode that is executable on the client specific device.
5. The method of Claim 4, wherein executing the application comprises configuring the configurable part of the client specific device; and thereafter executing the native software bytecode on the client specific device.
6. The method of Claim 1, wherein the client specific device comprises a programmable logic device.
7. The method of Claim 6, wherein the programmable logic device comprises a field programmable gate array.
8. The method of Claim 1, wherein the configurable part of the virtual device is modeled by a register transfer level description.
9. The method of Claim 8, wherein the register transfer level description is a description of a datapath that includes a netlist of register transfer level datapath cores and a controller that is described by microcode.
10. The method of Claim 9, wherein transforming comprises:

selecting a physical core implementation for the datapath cores within the datapath description; and

generating connections between the selected datapath core implementations from the netlist within the datapath description.

11. The method of Claim 10, wherein the datapath description provides a logic view for a plurality of physical core implementations, and wherein the datapath description and each of the physical core implementations have the same interface.

12. The method of Claim 1, wherein the virtual device comprises abstract logic blocks and an abstract routing architecture that comprises channel segments for connecting part of the abstract logic blocks, the method additionally comprising:

compiling the application and thereby generating abstract routing information;

and

indicating in an abstract coordinate system, which channel segments are used for connecting ports of the abstract logic blocks, the abstract routing information being part of the abstract bytecode.

13. The method of Claim 1, wherein the configurable part of the virtual device comprises abstract logic blocks and an abstract routing architecture that comprises channel segments for connecting part of the abstract logic blocks, wherein the abstract bytecode comprises abstract routing information that indicates in an abstract coordinate system, which channel segments are used for connecting ports of the abstract logic blocks, wherein the configurable part of the client specific device comprises local logic blocks and a local routing architecture for connecting part of the local logic blocks, and wherein transforming the abstract bytecode into native bytecode comprises mapping the abstract logic block into the local logic blocks.

14. The method of Claim 1, wherein transforming the abstract bytecode into native bytecode comprises:

generating an application programming interface description; and

generating bitstreams for reconfiguring the configurable part of the client specific device.

15. The method of Claim 14, wherein executing the application comprises compiling at least the application programming interface description and executing the compiled

application programming interface description, thereby generating the bitstreams for reconfiguring the configurable part of the client specific device.

16. The method of Claim 1, additionally comprising generating a virtual hardware/software interface that is representative of a class of hardware/software interfaces, the virtual hardware/software interface comprising a virtual hardware interface and virtual software interface.

17. The method of Claim 16, wherein executing the application on the client specific device comprises invoking a hardware/software interface, wherein the hardware/software interface comprising a virtual hardware/software interface and a local hardware/software interface that is specific for the client specific device.

18. The method of Claim 17, wherein the software bytecode communicates only with the virtual software interface, wherein the virtual software interface communicates with the local hardware/software interface, wherein the client specific device communicates only with the virtual hardware interface, and wherein the virtual hardware interface communicates with the local hardware/software interface.

19. A method of generating an abstract bytecode, the method comprising:  
compiling an application for execution on a virtual device thereby generating abstract bytecode, wherein the virtual device is at least partly configurable and is representative of a class of configurable devices that each includes a configurable client specific device, wherein the abstract bytecode is transformable into native bytecode for at least one of the configurable client specific devices, and wherein the native hardware bytecode comprises configuration information for the configurable client specific device.

20. The method of Claim 19, wherein the configurable client specific device is modeled at least in part by a register transfer level description.

21. A method of executing at least one client peer, having a client specific device, the method comprising:

receiving, at the client peer, abstract bytecode wherein the abstract bytecode contains abstract configuration information for a virtual device that is functionally representative of a class of devices, of which at least the client specific device is part;

transforming, at the client peer, the abstract bytecode into native bytecode for the client specific device, wherein the abstract configuration information is transformed into native configuration information;

configuring the client specific device using at least in part the native configuration information; and

executing the native bytecode at the client peer on the client specific device.

22. The method of Claim 21, wherein the abstract bytecode comprises abstract hardware bytecode that is transformable into native hardware bytecode that includes configuration information for the client specific device.

23. The method of Claim 21, wherein executing the native bytecode comprises configuring the configurable part of the client specific device, based on the native hardware bytecode.

24. The method of Claim 21, wherein the configurable part of the virtual device is modeled by a register transfer level description.

25. The method of Claim 21, wherein transforming the abstract bytecode into native bytecode, comprises:

generating an application programming interface description; and

generating and accessing bitstreams for reconfiguring the configurable part of the client specific device.

26. The method of Claim 24, wherein executing the native bytecode comprises:

compiling at least the application programming interface description; and

executing at least the compiled application programming interface description thereby generating the bitstreams for reconfiguring the configurable part of the client specific device.

27. A program storage device readable by a client specific device, tangibly embodying a program of instructions executable by the client specific device, to perform steps for executing an application on the client specific device, the method comprising:

transforming an abstract bytecode obtained by compilation of the application for execution on a virtual device that is at least partly configurable and is representative of a class of devices, of which at least the client specific device is part, into native bytecode

that is partly executable on the client specific device and that is partly used for configuring the client specific device.

28. The program storage device of Claim 27, wherein the abstract bytecode contains a register-transfer level description.

29. The program storage device of Claim 27, wherein the method further comprises:  
generating an application programming interface; and

compiling the application programming interface and executing at least the application programming interface, to thereby generate bitstreams for reconfiguring the configurable part of the client specific device.

30. A client specific device for executing an application, provided from a service peer, the device comprises:

a configurable hardware part;

means for receiving abstract bytecode, being obtainable by compilation of the application for execution on a virtual device that is at least partly configurable, wherein the virtual device is representative for a class of devices, of which at least the client specific device is part, wherein the configurable part of the virtual device being modeled by a register transfer level description;

means for transforming the abstract bytecode into native bytecode for the client specific device; and

means for executing the application/service at the client peer on the client specific device, being part of the class of devices and being at least partly configurable by exploiting the native bytecode.